

10/523786  
Rec'd CT/PTO 09 FEB 2005  
PCT/JP 03/10409

日 本 国 特 許 庁  
JAPAN PATENT OFFICE

18.08.03

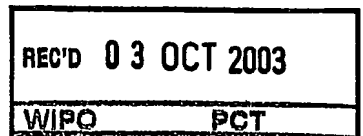
別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日  
Date of Application: 2 0 0 2 年 8 月 1 9 日

出 願 番 号  
Application Number: 特 願 2 0 0 2 - 2 3 8 4 3 7  
[ST. 10/C]: [ J P 2 0 0 2 - 2 3 8 4 3 7 ]

出 願 人  
Applicant(s): ソニー株式会社

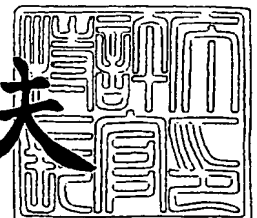


PRIORITY DOCUMENT  
SUBMITTED OR TRANSMITTED IN  
COMPLIANCE WITH  
RULE 17.1(a) OR (b)

2 0 0 3 年 9 月 1 9 日

特許庁長官  
Commissioner,  
Japan Patent Office

今 井 康 夫



BEST AVAILABLE COPY

【書類名】 特許願

【整理番号】 0290504806

【提出日】 平成14年 8月19日

【あて先】 特許庁長官 殿

【国際特許分類】 G06F 9/06  
G06F 13/14  
G06F 15/16

【発明者】

【住所又は居所】 東京都品川区北品川 6 丁目 7 番 3 5 号 ソニー株式会社  
内

【氏名】 杉森 茂夫

【特許出願人】

【識別番号】 000002185

【氏名又は名称】 ソニー株式会社

【代表者】 安藤 国威

【連絡先】 知的財産センター 0 3 - 5 4 4 8 - 2 1 3 7

【手数料の表示】

【予納台帳番号】 005094

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 情報処理方法とその方法を実現するプログラム及び記録媒体

【特許請求の範囲】

【請求項 1】 複数の中央演算処理装置と内部記憶手段とを含むプロセッサと、前記複数の中央演算処理装置において共通して実行される共通コード及び予め定められたいずれか一つの前記中央演算処理装置において実行される命令コードが格納された外部記憶手段と、ホスト演算手段とがバスにより相互接続されたシステムにおいて実現される情報処理方法であって、

前記ホスト演算手段の指示により、いずれか一つの前記中央演算処理装置が前記共通コードと自己において実行するよう定められた前記命令コードとを前記外部記憶手段から前記内部記憶手段へロードすると共に、他の前記中央演算処理装置が自己において実行するよう定められた前記命令コードを前記外部記憶手段から前記内部記憶手段へロードするステップと、

各々の前記中央演算処理装置が、前記内部記憶手段にロードされた前記共通コードと前記自己において実行するよう定められた命令コードとをそれぞれ実行するステップとを有することを特徴とする情報処理方法。

【請求項 2】 前記ホスト演算手段が選択的に前記中央演算処理装置をリセットするステップと、

前記ホスト演算手段の指示により、選択的にリセットされた前記中央演算処理装置が前記自己において実行するよう定められた命令コードを前記外部記憶手段から前記内部記憶手段へ新たにロードするステップと、

リセットされた前記中央演算処理装置が、前記内部記憶手段に新たにロードされた前記自己において実行するよう定められた命令コードを実行するステップとをさらに有する請求項 1 に記載の情報処理方法。

【請求項 3】 複数の中央演算処理装置と内部記憶手段とブート用記憶手段とダイレクトメモリアクセスコントローラを含むプロセッサと、前記複数の中央演算処理装置において共通して実行される共通コード及び予め定められたいずれか一つの前記中央演算処理装置において実行される命令コードが格納された外部記憶手段と、ホスト演算手段とがバスにより相互接続されたシステムにおいて実

現される情報処理方法であって、

前記ホスト演算手段が前記中央演算処理装置を選択的にリセットするステップと、

前記ホスト演算手段が、リセットされた前記中央演算処理装置により実行される起動コードを前記ブート用記憶手段へ書き込むステップと、

前記ホスト演算手段が、前記リセットされた中央演算処理装置のリセット状態を解除するステップと、

リセット状態が解除されたいずれか一つの前記中央演算処理装置が前記ブート用記憶手段に書き込まれた前記起動コードを実行することにより、前記共通コードと自己において実行するよう定められた前記命令コードとを前記ダイレクトメモリアクセスコントローラを用いて前記外部記憶手段から前記内部記憶手段へロードすると共に、リセット状態が解除された他の前記中央演算処理装置が前記ブート用記憶手段に書き込まれた前記起動コードを実行することにより、自己において実行するよう定められた前記命令コードを前記ダイレクトメモリアクセスコントローラを用いて前記外部記憶手段から前記内部記憶手段へロードするステップと、

リセット状態が解除された各々の前記中央演算処理装置が、前記内部記憶手段にロードされた前記共通コードと前記自己において実行するよう定められた命令コードとをそれぞれ実行するステップとを有することを特徴とする情報処理方法。

【請求項 4】 前記ホスト演算手段が選択的に前記中央演算処理装置を新たにリセットするステップと、

前記ホスト演算手段が、新たにリセットされた前記中央演算処理装置により実行される新たな起動コードを前記ブート用記憶手段へ書き込むステップと、

前記ホスト演算手段が、前記新たにリセットされた前記中央演算処理装置のリセット状態を解除するステップと、

リセット状態が解除された前記中央演算処理装置が前記新たな起動コードを実行することにより、前記自己において実行するよう定められた命令コードを前記ダイレクトメモリアクセスコントローラを用いて前記外部記憶手段から前記内部

記憶手段へ新たにロードするステップと、

前記リセット状態が解除された中央演算処理装置が、前記内部記憶手段に新たにロードされた前記自己において実行するよう定められた命令コードを実行するステップとをさらに有する請求項 3 に記載の情報処理方法。

【請求項 5】 複数の中央演算処理装置と内部記憶手段とを含むプロセッサと、前記複数の中央演算処理装置において共通して実行される共通コード及び予め定められたいずれか一つの前記中央演算処理装置において実行される命令コードが格納された外部記憶手段と、ホスト演算手段とがバスにより相互接続されたシステムにおいて実行されるプログラムであって、

いずれか一つの前記中央演算処理装置に対し、前記ホスト演算手段からの指示に応じて前記共通コードと自己において実行するよう定められた前記命令コードとを前記外部記憶手段から前記内部記憶手段へロードさせ、他の前記中央演算処理装置に対し、自己において実行するよう定められた前記命令コードを前記外部記憶手段から前記内部記憶手段へロードさせ、

各々の前記中央演算処理装置に対して、前記内部記憶手段にロードされた前記共通コードと前記自己において実行するよう定められた命令コードとをそれぞれ実行させることを特徴とするプログラム。

【請求項 6】 さらに、前記ホスト演算手段に対して、選択的に前記中央演算処理装置をリセットさせ、

選択的にリセットされた前記中央演算処理装置に対し、前記ホスト演算手段の指示により、前記自己において実行するよう定められた命令コードを前記外部記憶手段から前記内部記憶手段へ新たにロードさせ、

リセットされた前記中央演算処理装置に対して、前記内部記憶手段に新たにロードされた前記自己において実行するよう定められた命令コードを実行させる請求項 5 に記載のプログラム。

【請求項 7】 複数の中央演算処理装置と内部記憶手段とブート用記憶手段とダイレクトメモリアクセスコントローラを含むプロセッサと、前記複数の中央演算処理装置において共通して実行される共通コード及び予め定められたいずれか一つの前記中央演算処理装置において実行される命令コードが格納された外部

記憶手段と、ホスト演算手段とがバスにより相互接続されたシステムにおいて実行されるプログラムであって、

前記ホスト演算手段に対して、前記中央演算処理装置を選択的にリセットさせ

前記ホスト演算手段に対して、リセットされた前記中央演算処理装置により実行される起動コードを前記ブート用記憶手段へ書き込ませ、

前記ホスト演算手段に対して、前記リセットされた中央演算処理装置のリセット状態を解除させ、

リセット状態が解除されたいずれか一つの前記中央演算処理装置に対して、前記ブート用記憶手段に書き込まれた前記起動コードを実行させることにより、前記共通コードと自己において実行するよう定められた前記命令コードとを前記ダイレクトメモリアクセスコントローラを用いて前記外部記憶手段から前記内部記憶手段へロードさせると共に、リセット状態が解除された他の前記中央演算処理装置に対して、前記ブート用記憶手段に書き込まれた前記起動コードを実行させることにより、自己において実行するよう定められた前記命令コードを前記ダイレクトメモリアクセスコントローラを用いて前記外部記憶手段から前記内部記憶手段へロードさせ、

リセット状態が解除された各々の前記中央演算処理装置に対して、前記内部記憶手段にロードされた前記共通コードと前記自己において実行するよう定められた命令コードとをそれぞれ実行させることを特徴とするプログラム。

【請求項 8】 さらに、前記ホスト演算手段に対して選択的に前記中央演算処理装置を新たにリセットさせ、

前記ホスト演算手段に対して、新たにリセットされた前記中央演算処理装置により実行される新たな起動コードを前記ブート用記憶手段へ書き込ませ、

前記ホスト演算手段に対して、前記新たにリセットされた中央演算処理装置のリセット状態を解除させ、

リセット状態が解除された前記中央演算処理装置に対して、前記新たな起動コードを実行させることにより、前記自己において実行するよう定められた命令コードを前記ダイレクトメモリアクセスコントローラを用いて前記外部記憶手段か

ら前記内部記憶手段へ新たにロードさせ、

前記リセット状態が解除された中央演算処理装置に対して、前記内部記憶手段に新たにロードされた前記自己において実行するよう定められた命令コードを実行させる請求項 7 に記載のプログラム。

【請求項 9】 複数の中央演算処理装置と内部記憶手段とを含むプロセッサと、前記複数の中央演算処理装置において共通して実行される共通コード及び予め定められたいずれか一つの前記中央演算処理装置において実行される命令コードが格納された外部記憶手段と、ホスト演算手段とがバスにより相互接続されたシステムにおいて実行されるプログラムを記録した記録媒体であって、前記プログラムは、

いずれか一つの前記中央演算処理装置に対し、前記ホスト演算手段からの指示に応じて前記共通コードと自己において実行するよう定められた前記命令コードとを前記外部記憶手段から前記内部記憶手段へロードさせ、他の前記中央演算処理装置に対し、自己において実行するよう定められた前記命令コードを前記外部記憶手段から前記内部記憶手段へロードさせ、

各々の前記中央演算処理装置に対して、前記内部記憶手段にロードされた前記共通コードと前記自己において実行するよう定められた命令コードとをそれぞれ実行させることを特徴とする記録媒体。

【請求項 10】 前記プログラムはさらに、

前記ホスト演算手段に対して、選択的に前記中央演算処理装置をリセットさせ、

選択的にリセットされた前記中央演算処理装置に対し、前記ホスト演算手段の指示により、前記自己において実行するよう定められた命令コードを前記外部記憶手段から前記内部記憶手段へ新たにロードさせ、

リセットされた前記中央演算処理装置に対して、前記内部記憶手段に新たにロードされた前記自己において実行するよう定められた命令コードを実行させる請求項 9 に記載の記録媒体。

【請求項 11】 複数の中央演算処理装置と内部記憶手段とブート用記憶手段とダイレクトメモリアクセスコントローラを含むプロセッサと、前記複数の中

中央演算処理装置において共通して実行される共通コード及び予め定められたいずれか一つの前記中央演算処理装置において実行される命令コードが格納された外部記憶手段と、ホスト演算手段とがバスにより相互接続されたシステムにおいて実行されるプログラムを記録した記録媒体であって、前記プログラムは、

前記ホスト演算手段に対して前記中央演算処理装置を選択的にリセットさせ、

前記ホスト演算手段に対して、リセットされた前記中央演算処理装置により実行される起動コードを前記ブート用記憶手段へ書き込ませ、

前記ホスト演算手段に対して、前記リセットされた中央演算処理装置のリセット状態を解除させ、

リセット状態が解除されたいずれか一つの前記中央演算処理装置に対して、前記ブート用記憶手段に書き込まれた前記起動コードを実行させることにより、前記共通コードと自己において実行するよう定められた前記命令コードとを前記ダイレクトメモリアクセスコントローラを用いて前記外部記憶手段から前記内部記憶手段へロードさせると共に、リセット状態が解除された他の前記中央演算処理装置に対して、前記ブート用記憶手段に書き込まれた前記起動コードを実行させることにより、自己において実行するよう定められた前記命令コードを前記ダイレクトメモリアクセスコントローラを用いて前記外部記憶手段から前記内部記憶手段へロードさせ、

リセット状態が解除された各々の前記中央演算処理装置に対して、前記内部記憶手段にロードされた前記共通コードと前記自己において実行するよう定められた命令コードとをそれぞれ実行させることを特徴とする記録媒体。

【請求項 12】 前記プログラムはさらに、

前記ホスト演算手段に対して選択的に前記中央演算処理装置を新たにリセットさせ、

前記ホスト演算手段に対して、新たにリセットされた前記中央演算処理装置により実行される新たな起動コードを前記ブート用記憶手段へ書き込ませ、

前記ホスト演算手段に対して、前記新たにリセットされた中央演算処理装置のリセット状態を解除させ、

リセット状態が解除された前記中央演算処理装置に対して、前記新たな起動コ



ードを実行させることにより、前記自己において実行するよう定められた命令コードを前記ダイレクトメモリアクセスコントローラを用いて前記外部記憶手段から前記内部記憶手段へ新たにロードさせ、

前記リセット状態が解除された中央演算処理装置に対して、前記内部記憶手段に新たにロードされた前記自己において実行するよう定められた命令コードを実行させる請求項 11 に記載の記録媒体。

#### 【発明の詳細な説明】

##### 【0001】

#### 【発明の属する技術分野】

本発明は情報処理方法とその方法を実現するためのプログラム、及び該プログラムを記録した記録媒体に関するものである。

##### 【0002】

#### 【従来の技術】

近年においては、一つの半導体チップ上にCPU（中央演算処理装置）やメモリが搭載されたシステムオンチップ（以下、単に「SOC」ともいう）が開発されてきている。このようなSOCは、CPUとメモリとの間におけるバス幅が広く取れるという利点を有し、構成要素としてシステムに組み込まれる。ここでSOCにおいては、該チップの大きさに応じて混載できるメモリの容量に限度があるため、混載されたメモリを効率的に使用することが重要とされる。

##### 【0003】

一方、近年における音楽配信や携帯オーディオ機器に使用されている音声圧縮方式（コーデック-Codec）には、MPEG1 Audio、MPEG2 Audio等多くの種類が存在する。このとき、音声データの圧縮・伸長を実行するシステムを複数のCPU（マルチプロセッサ）により構成することによって負荷を分散し、一つのCPUでは時間のかかる処理を高速処理することができる。すなわち例えば、コーデック毎にCPUを割り当てることにより、第1のCPUによりデコードしたデータを同時に第2のCPUでエンコードするトランスコーディング動作や、異なるコーデックで並列的にエンコードする動作を実現することができる。

##### 【0004】

図5は、従来の情報処理システムの構成を示すブロック図である。図5に示されるように、従来の情報処理システムはバス3と、バス3により相互接続されるチップ1と外部メモリ110とホストCPU111及びサーバ120を備える。そして、チップ1には内部メモリ101と、外部メモリ110から直接外部メモリ110へ実行可能コードやデータを転送するDMA (Direct Memory Access) コントローラ102、第1CPU103、第2CPU104、及びブート用メモリ105を含む。なお、チップ1内に形成された内部メモリ101とDMA (Direct Memory Access) コントローラ102、第1CPU103、第2CPU104、及びブート用メモリ105は、チップ1内に形成されたバスにより相互接続される。

#### 【0005】

ここで、上記のような構成を有するチップ1は、相互に独立して動作すると共に、内部メモリ101を共有する複数のCPUを備えるため、「粗結合マルチプロセッサ」と呼ばれる。

#### 【0006】

上記のような構成を有する情報処理システムでは、まずホストCPU111の指示により第1CPU103用及び第2CPU104用のブートストラップがブート用メモリ105に格納される。そして、第1CPU103及び第2CPU104が、上記ブートストラップに応じて外部メモリ110又はネットワーク上のサーバ120からDMAコントローラ102を使用して実行可能コードを内部メモリ101へダウンロードし、該システムを起動する。

#### 【0007】

一方、上記実行可能コードは以下のように作成される。図6に示されるように、チップ1内に設けられたCPU毎に作成されたプログラム（第1CPU用プログラム及び第2CPU用プログラム）と、複数のCPUにより共用されるプログラム（共通ライブラリプログラム）とをコンパイルさせることにより、各プログラムに対応したオブジェクトコード（CPU0用オブジェクトコード、CPU1用オブジェクトコード、共通ライブラリオブジェクトコード）が生成される。

#### 【0008】

次に、これらのオブジェクトコードは、内部メモリ 101 内における配置先を指定する先頭アドレスを含んだリンク情報 505 とリンクされ、実行可能コードが生成される。従って、生成された実行可能コード 506 は、命令とデータ、及び内部メモリ 101 への配置先アドレスを記述するものとされる。なお、図 6 においては、第 1 CPU に対しコーデック A においてエンコード動作を実現させ、第 2 CPU に対しコーデック B においてデコード動作を実現させるための実行可能コード 506 が例示される。

#### 【0009】

そして、後述するように該実行可能コード 506 は内部メモリ 101 にロードされる。

#### 【0010】

ここで、該情報システムの OS (Operating System) がダイナミックライブラリやリンク機能を有し、あるいはハードウェア的に仮想アドレスをサポートするものである場合には、一般的にプログラムのロード時に実行可能コードのアドレスを変更することができるものの、上記のような機能を持たない OS が搭載されたシステム、または OS が搭載されていないシステムに該チップ 1 が組み込まれるような場合には実行可能コード作成時にアドレスが固定され、動的にコードのロード先を切り替えることができない。

#### 【0011】

従って、複数のコーデックにより CPU を動作させるためには、想定される全パターンのコードを内部メモリ 101 等に保持しておく必要がある。このとき、使用したいコーデックが多数あって内部メモリ 101 内に該コードが納まらない場合には、該実行可能コードを外部メモリ 110 やサーバ 120 等に保持しておき、必要に応じたダウンロードを実行してコーデックの種類や動作を切り替えることができる。

#### 【0012】

ただし、図 6 に示されるように、従来の実行可能コード 506 内には第 1 CPU 用のコードと第 2 CPU 用のコードとがまとめられているため、第 1 CPU 103 あるいは第 2 CPU 104 のいずれか一方の動作のみを切り替えたいような

場合には、実行可能コード506全体を内部メモリ101にロードし直す必要がある。

#### 【0013】

ここで、図7及び図8を参照して、従来の情報処理システムにおける該動作の切り替えを説明する。なお、図7においては外部メモリ110に第1から第4の命令コード212～215が予め格納され、最初に第1命令コード212が内部メモリ101にロードされた場合が示される。上記において第2命令コード213は第1CPU103に対してコーデックAによりエンコード動作させ、第2CPU104に対してコーデックDによりデコード動作させるコードを含み、第3命令コード214は第1CPU103に対してコーデックCによりエンコード動作させ、第2CPU104に対してコーデックBによりデコード動作させるコードを含み、第4命令コード215は第1CPU103に対してコーデックCによりエンコード動作させ、第2CPU104に対してコーデックDによりデコード動作させるコードを含むものとされる。

#### 【0014】

図8を参照して、図7に示された第1及び第2のCPU103, 104の動作を説明する。まずステップS1においてホストCPU111が第1CPU103をリセットし、ステップS2においてホストCPU111がブート用メモリ105にブートストラップを書き込む。次に、ステップS3において、ホストCPU111が第1CPU103のリセット状態を解除する。そして、ステップS4において、第1CPU103はブート用メモリ105に書き込まれたブートストラップを実行し、ステップS5においてDMAコントローラ102を用いて外部メモリ110から内部メモリ101へ例えば第1命令コード212をDMA転送する。

#### 【0015】

ステップS6では、第1CPU103が上記転送の終了を確認した後に第2CPU104をリセットし、さらに該リセット状態を解除することにより第2CPU104を起動する。ステップS7では、第1CPU103は内部メモリ101に格納された第1CPU103用の命令コードを実行する。

## 【0016】

これより、ステップS8において第1CPU103はコーデックAのエンコーダとして動作すると共に、ステップS9において第2CPU104は内部メモリ101に格納された第2CPU104用の命令コードを実行することによってコーデックBのデコーダとして動作する。

## 【0017】

このとき上記のように、第1CPU用の命令コードと第2CPU用の命令コードとが一つの命令コードとしてまとめられていることから、例えば第1CPU103をコーデックCのエンコーダとして動作させたい場合には、たとえ第2CPU104の機能を変更する必要がある場合であっても、内部メモリ101にロードされた第1命令コード212全体を第3命令コード214に入れ替える必要がある。

## 【0018】

以上より、上記のような従来の情報処理システムにおいては、以下のような問題がある。まず、第1CPU103向けの命令と第2CPU104向けの命令をまとめてコンパイルするため、実行可能コードを構成する命令の組み合わせが固定的となってしまう。このため、複数のCPUを複数のコーデックにおいて動作させるためには、各動作状態に応じたコンパイル済みコードを予め外部メモリ110（あるいはネットワーク上のサーバ120）に保持しておく必要がある。

## 【0019】

また、上記のように第1CPU103向けの命令と第2CPU104向けの命令とにより一つの実行可能コードが構成されるため、実行可能コードのサイズが大きくなる。これより、入れ替えるべき内部メモリ101内のコードサイズも大きなものとなるため、コード入れ替え時間が大きなものとなる。

## 【0020】

さらに、第1CPU103向けの命令と第2CPU104向けの命令とにより一つの実行可能コードが構成されることから、上記のように例えば第1CPU103の動作のみを変更したい場合でも、第2CPU104の動作も中断させなければならない。すなわち例えば、第2CPU104によりコーデックBの下でデ

コードした結果を、第 1 C P U 1 0 3 によりコーデック A の下でエンコードする  
トランスコーディング動作中においては、第 1 C P U 1 0 3 の動作のみを変更する  
ことはできず、第 2 C P U 1 0 4 のデコーダも中断させなくてはならない。

#### 【 0 0 2 1 】

##### 【発明が解決しようとする課題】

本発明は、上記のような問題を解消するためになされたもので、複数の C P U  
を有する情報処理システムにおいて、該システムにおいて必要とされる記憶容量  
を低減できると共に処理速度を高速化し、他の C P U の動作に影響を与えること  
なく各 C P U の機能を容易に変更することのできる情報処理方法と、その方法を  
実現するプログラム及び該プログラムを記録した記録媒体を提供することを目的  
とする。

#### 【 0 0 2 2 】

##### 【課題を解決するための手段】

本発明の目的は、複数の中央演算処理装置と内部記憶手段とを含むプロセッサ  
と、複数の中央演算処理装置において共通して実行される共通コード及び予め定  
められたいずれか一つの中央演算処理装置において実行される命令コードが格納  
された外部記憶手段と、ホスト演算手段とがバスにより相互接続されたシステム  
において実現される情報処理方法であって、ホスト演算手段の指示により、いず  
れか一つの中央演算処理装置が共通コードと自己において実行するよう定められ  
た命令コードとを外部記憶手段から内部記憶手段へロードすると共に、他の中央  
演算処理装置が自己において実行するよう定められた命令コードを外部記憶手段  
から内部記憶手段へロードするステップと、各々の中央演算処理装置が、内部記  
憶手段にロードされた共通コードと自己において実行するよう定められた命令コ  
ードとをそれぞれ実行するステップとを有することを特徴とする情報処理方法、  
あるいは該方法を実現するためのプログラム若しくは該プログラムを記録した記  
録媒体を提供することにより達成される。

#### 【 0 0 2 3 】

このような手段によれば、各々の中央演算処理装置が自己において実行するよ  
う定められた命令コードを内部記憶手段へロードして実行するため、外部記憶手

段へ格納すべき共通コード及び命令コードを効率化することができ、外部記憶手段に必要とされる記憶容量と、外部記憶手段から内部記憶手段にロードすべき情報量とを低減することができる。

#### 【0024】

また、上記において、ホスト演算手段が選択的に中央演算処理装置をリセットするステップと、ホスト演算手段の指示により、選択的にリセットされた中央演算処理装置が自己において実行するよう定められた命令コードを外部記憶手段から内部記憶手段へ新たにロードするステップと、リセットされた中央演算処理装置が、内部記憶手段に新たにロードされた自己において実行するよう定められた命令コードを実行するステップとをさらに有する情報処理方法、あるいは該方法を実現するためのプログラム若しくは該プログラムを記録した記録媒体によれば、他の中央演算処理装置の動作に影響を与えることなく、選択された中央演算処理装置の機能を容易に変更することができる。

#### 【0025】

##### 【発明の実施の形態】

以下において、本発明の実施の形態を図面を参照しつつ詳しく説明する。なお、図中同一符号は同一または相当部分を示す。

#### 【0026】

図1は、本発明の実施の形態に係る情報処理を説明する図である。図1に示されるように、本実施の形態に係る情報処理は、バス3により相互接続されたホストCPU411とチップ1、及び外部メモリ110により構成された情報処理システムにおいて実現される。ここで、チップ1には内部バスにより相互接続されたDMAコントローラ102と第1CPU103、第2CPU104、ブート用メモリ105、及び内部メモリ101が設けられる。

#### 【0027】

なお、上記において外部メモリ110の替わりに、あるいは外部メモリ110と共に、バス3に接続されたネットワーク上のサーバを備えた情報システムにおいても本実施の形態に係る情報処理を実現することができる。

#### 【0028】

以下において、図2を参照しつつ、上記情報処理システムにおいて実行される本実施の形態に係る実行可能コードについて説明する。本実施の形態においては、上記第1CPU103と第2CPU104が独立して動作できるように、各CPUで実行されるオブジェクトコードにそれぞれ共通オブジェクトコードがリンクされ、CPU毎の実行可能コードが生成される。

#### 【0029】

すなわち、図2に示されるように、第1CPU103にて実行される第1CPU用プログラムがコンパイルされることにより第1CPU用オブジェクトコードが生成され、第2CPU104にて実行される第2CPU用プログラムがコンパイルされることにより第2CPU用オブジェクトコードが生成される。また、第1及び第2CPU103、104にて実行される共通ライブラリプログラムがコンパイルされることにより共通ライブラリオブジェクトコードが生成される。

#### 【0030】

次に、内部メモリ101内における共通オブジェクトコードの格納先を示す共通ライブラリ先頭アドレスと、内部メモリ101内における第1CPUオブジェクトコードの格納先を示す第1CPU先頭アドレスとを記述したリンク情報302を上記第1CPU用オブジェクトコードと上記共通ライブラリオブジェクトコードにリンクさせて実行可能コード305を生成すると共に、上記と同じ共通ライブラリ先頭アドレスと、内部メモリ101内における第2CPUオブジェクトコードの格納先を示す第2CPU先頭アドレスとを記述したリンク情報304を上記第2CPU用オブジェクトコードと上記共通ライブラリオブジェクトコードにリンクさせて実行可能コード306を生成する。

#### 【0031】

なお、上記の各先頭アドレスは、対応する各オブジェクトコードが内部メモリ101に格納される際に必要とされる記憶容量を考慮して決定される。

#### 【0032】

また、上記において共通オブジェクトコードが記述される共通ライブラリ領域においては読み出し専用のテキスト命令が作成され、第1CPU103用あるいは第2CPU104用のオブジェクトコードが記述される領域においては読み出



し専用のテキスト命令と読み書き可能なデータ領域とが作成される。

#### 【0033】

ここで、オブジェクトコード301及びオブジェクトコード303に含まれる共通オブジェクトコードは全く同じものとされ、リンク情報302、304に含まれる該共通オブジェクトコードに対応した先頭アドレスも同じアドレスとされる。従って、上記実行可能コード305、306が共に内部メモリ101へロードされた場合には、図2に示されるように共通オブジェクトコードが共有される。

#### 【0034】

これより、本実施の形態に係る情報処理においては、後述するように第1CPU103と第2CPU104とにおいて一つの共通オブジェクトコードが共用されるため、必要とされる内部メモリ101の容量が低減される。

#### 【0035】

次に、図1を参照しつつ、上記情報処理システムにおいて実現される上記実行可能コード305、306を用いた情報処理方法を説明する。

#### 【0036】

図1に示されるように、外部メモリ110には予め共通ライブラリ412と第1CPUコード群CG0、及び第2CPUコード群CG1の3種類の実行可能ファイルが格納される。ここで、第1CPUコード群CG0には第1CPU103をコーデックAのエンコーダとして動作させる命令コード413と第1CPU103をコーデックCのエンコーダとして動作させる命令コード415とが含まれる。また、第2CPUコード群CG1には第2CPU104をコーデックBのデコーダとして動作させる命令コード414と第2CPU103をコーデックDのデコーダとして動作させる命令コード416とが含まれる。

#### 【0037】

なお、共通ライブラリ412は第1CPU103と第2CPU104とにおいて共用される命令と、該命令の内部メモリ101内における配置先を指定するアドレスとを含み、先頭アドレスが予め決められる。また、命令コード413、415は、第1CPU103に対する命令とデータ及びそれらの内部メモリ101

内における配置先を示すアドレスを含む。同様に命令コード 414, 416 は、第 2 CPU 104 に対する命令とデータ及びそれらの内部メモリ 101 内における配置先を示すアドレスを含む。

#### 【0038】

そして、初回起動時には、共通ライブラリ 412 と第 1 CPU 用の命令コード 413 及び第 2 CPU 用の命令コード 414 が、ブートストラップを使って内部メモリ 101 にロードされる。このとき、共通ライブラリ 412 は第 1 CPU 103 と第 2 CPU 104 とにより共用されるので、一度だけ内部メモリ 101 にロードされれば足りる。

#### 【0039】

また、本実施の形態においては、内部メモリ 101 の記憶領域が第 1 領域 R1 と第 2 領域 R2 及び第 3 領域 R3 に予め分割され、上記共通ライブラリ 412 は第 1 領域 R1 に格納され、第 1 CPU 用の命令コード 413, 415 は第 2 領域 R2 に格納され、第 2 CPU 用の命令コード 414, 416 は第 3 領域 R3 に格納される。

#### 【0040】

以下において、図 3 を参照しつつ、本実施の形態に係る情報処理をより詳しく説明する。ステップ S1 においてホスト CPU 411 が第 1 CPU 103 をリセットし、ステップ S2 においてホスト CPU 411 が第 2 CPU 104 をリセットする。そして、ステップ S3 では、ホスト CPU 411 がブート用メモリ 105 に第 1 CPU 用のブートストラップを書き込み、ステップ S4 においてホスト CPU 411 が同じブート用メモリ 105 に第 2 CPU 用のブートストラップを書き込む。

#### 【0041】

ステップ S5 においてはホスト CPU 411 が第 1 CPU 103 のリセット状態を解除し、ステップ S6 においてはホスト CPU 411 が第 2 CPU 104 のリセット状態を解除する。

#### 【0042】

ステップ S7 では、ホスト CPU 411 の指示により第 1 CPU 103 がブー

ト用メモリ105に格納された第1CPU用のブートストラップを実行し、第1CPU103がDMAコントローラ102を用いて外部メモリ110から内部メモリ101へ共通ライブラリ412と第1CPU用の命令コード413とをDMA転送する。

【0043】

次に、ステップS8においては、ホストCPU411の指示により第2CPU104がブート用メモリ105に格納された第2CPU用のブートストラップを実行し、第2CPU104がDMAコントローラ102を用いて外部メモリ110から内部メモリ101へ第2CPU用の命令コード414をDMA転送する。

【0044】

そして、ステップS9において、第1CPU103は命令コード413の該DMA転送が終了したことを確認した後に、内部メモリ101に格納された第1CPU用の命令コード413を実行する。また、ステップS10において、第2CPU104は命令コード414の該DMA転送が終了したことを確認した後に、内部メモリ101に格納された第2CPU用の命令コード414を実行する。

【0045】

なお、上記ステップS9及びステップS10において、第1CPU103と第2CPU104は必要に応じて内部メモリ101に格納された共通ライブラリ412を読み出して実行する。

【0046】

これより、ステップS11において第1CPU103はコーデックAのエンコーダとして動作を開始し、ステップS12において第2CPU104はコーデックBのデコーダとして動作を開始する。

【0047】

次に、図1及び図4を参照して、第1CPU103あるいは第2CPU104の機能を変更する場合の情報処理方法を説明する。本実施の形態に係る情報処理方法においては、少なくとも第1CPU103又は第2CPU104のいずれか一方の機能を変更する場合には、内部メモリ101にロードされている第1CPU用あるいは第2CPU用の命令コードを入れ替える必要がある。

## 【0048】

従って、第1CPU103の機能を変更する場合には、第1CPUコード群CG0に含まれた第1CPU用の他の命令コードを内部メモリ101にロードし、第2CPU104の機能を変更する場合には、第2CPUコード群CG1に含まれた第2CPU用の他の命令コードを内部メモリ101にロードして実行する。

## 【0049】

このとき、各CPUは共通ライブラリ412に記述されたコードを読み取ると共に、内部メモリ101における該CPU専用のデータ領域においてデータの読み書きを実行しているため、上記のような命令コードの入れ替えは他のCPUの動作に影響を与えることがない。

## 【0050】

以下において、図4を参照しつつ、図1に示された第1CPU103に対する機能変更方法の具体例を詳しく説明する。

## 【0051】

まずステップS1において、上記のように第1CPU103はコーデックAでデータをエンコードし、第2CPU104は上記のように第1CPU103の動作によりエンコードされたデータをコーデックBでデコードしているものとする。

## 【0052】

次に、ステップS2においてホストCPU411が第1CPU103をリセットする。そして、ステップS3において、ホストCPU411がブート用メモリ105に第1CPU用のブートストラップを書き込む。

## 【0053】

ステップS4において、ホストCPU411が第1CPU103のリセット状態を解除する。さらに、ステップS5において第1CPU103はブート用メモリ105に書き込まれたブートストラップを実行し、第1CPU103はDMAコントローラ102を用いて外部メモリ110から内部メモリ101へ第1CPU用の命令コード415のみをDMA転送する。

## 【0054】

次に、ステップ S 6 において、第 1 CPU 103 は命令コード 415 の該 DMA 転送が終了したことを確認した後に、内部メモリ 101 に格納された命令コード 415 を実行する。これより、第 1 CPU 103 はその機能が変更され、ステップ S 7 においてコーデック C のエンコーダとして動作する。

#### 【0055】

以上より、本発明の実施の形態に係る情報処理方法においては、外部メモリ 110 あるいはサーバ 120 に格納される共通ライブラリ 412 が第 1 CPU 103 と第 2 CPU 104 とにおいて共用される。また、各 CPU により実行されるオブジェクトコードが独立して外部メモリ 110 等に格納され、該オブジェクトコードを単位として内部メモリ 101 へロードされる。

#### 【0056】

これより、本発明の実施の形態に係る情報処理方法によれば、他の CPU の動作に影響を与えることなく各 CPU の機能を変更することができると共に、外部メモリ 110 等に保持すべきコードのサイズを低減することができる。

#### 【0057】

また、本発明の実施の形態に係る情報処理方法によれば、上記機能の変更に際して入れ替えるべき内部メモリ 101 内の命令コードのサイズが低減されるため、高速に CPU の機能を変更することができる。

#### 【0058】

なお、上記情報処理方法はコンピュータにより実行されるプログラムにより記述でき、さらに該プログラムはフレキシブルディスクや CD-ROM 等の記録媒体に記録することができる。従って、該プログラムをコンピュータにより実行することにより、本実施の形態に係る情報処理方法を容易に実現することができる。

#### 【0059】

以上より、本発明の実施の形態に係る情報処理方法によれば、複数のコーデックによる音声圧縮展開プログラムを外部メモリ 110 に効率的に保持することができると共に、ある CPU に実行させるプログラム（命令コード）を他の CPU の動作に影響を与えることなく入れ替えることができる。

## 【0060】

## 【発明の効果】

本発明に係る情報処理方法あるいは該方法を実現するためのプログラム若しくは該プログラムを記録した記録媒体によれば、外部記憶手段に必要とされる記憶容量と、外部記憶手段から内部記憶手段にロードすべき情報量とを低減することができるため、該システム規模及びコストを低減し、該システムにおける情報処理速度を高めることができる。

## 【0061】

また、本発明に係る情報処理方法あるいは該方法を実現するためのプログラム若しくは該プログラムを記録した記録媒体によれば、他の中央演算処理装置の動作に影響を与えることなく、選択された中央演算処理装置の機能を容易に変更することができるため、動作の信頼性を担保しつつ該システムの汎用性を高めることができる。

## 【図面の簡単な説明】

## 【図1】

本発明の実施の形態に係る情報処理を説明する図である。

## 【図2】

本発明の実施の形態に係る実行可能コードを説明する図である。

## 【図3】

図1に示された第1CPU及び第2CPUの動作を示すフローチャートである。

## 【図4】

図1に示された第1CPUに対する機能変更方法の具体例を示すフローチャートである。

## 【図5】

従来の情報処理システムの構成を示すブロック図である。

## 【図6】

従来の実行可能コードを説明する図である。

## 【図7】

図 5 に示された情報処理システムの動作を説明する図である。

【図 8】

図 5 に示された第 1 CPU 及び第 2 CPU の動作を示すフローチャートである。

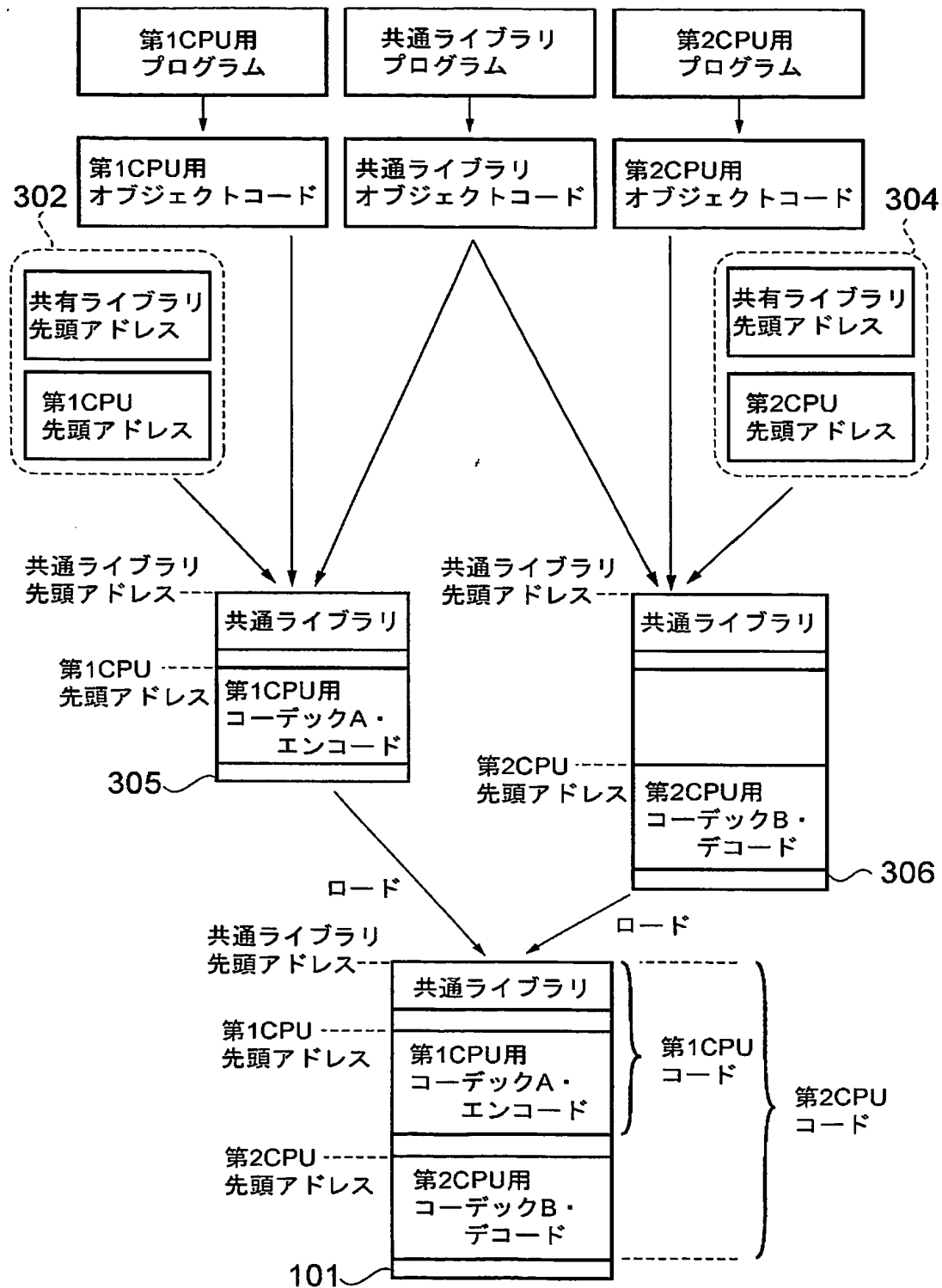
【符号の説明】

1 チップ、3 バス、101 内部メモリ、102 DMA コントローラ、103 第 1 CPU、104 第 2 CPU、105 ブート用メモリ、110 外部メモリ、111, 411 ホスト CPU、120 サーバ、212 第 1 命令コード、213 第 2 命令コード、214 第 3 命令コード、215 第 4 命令コード、302, 304, 505 リンク情報、305, 306, 506 実行可能コード、412 共通ライブラリ、413～416 命令コード、501 オブジェクトコード、CG0 第 1 CPU コード群、CG1 第 2 CPU コード群。

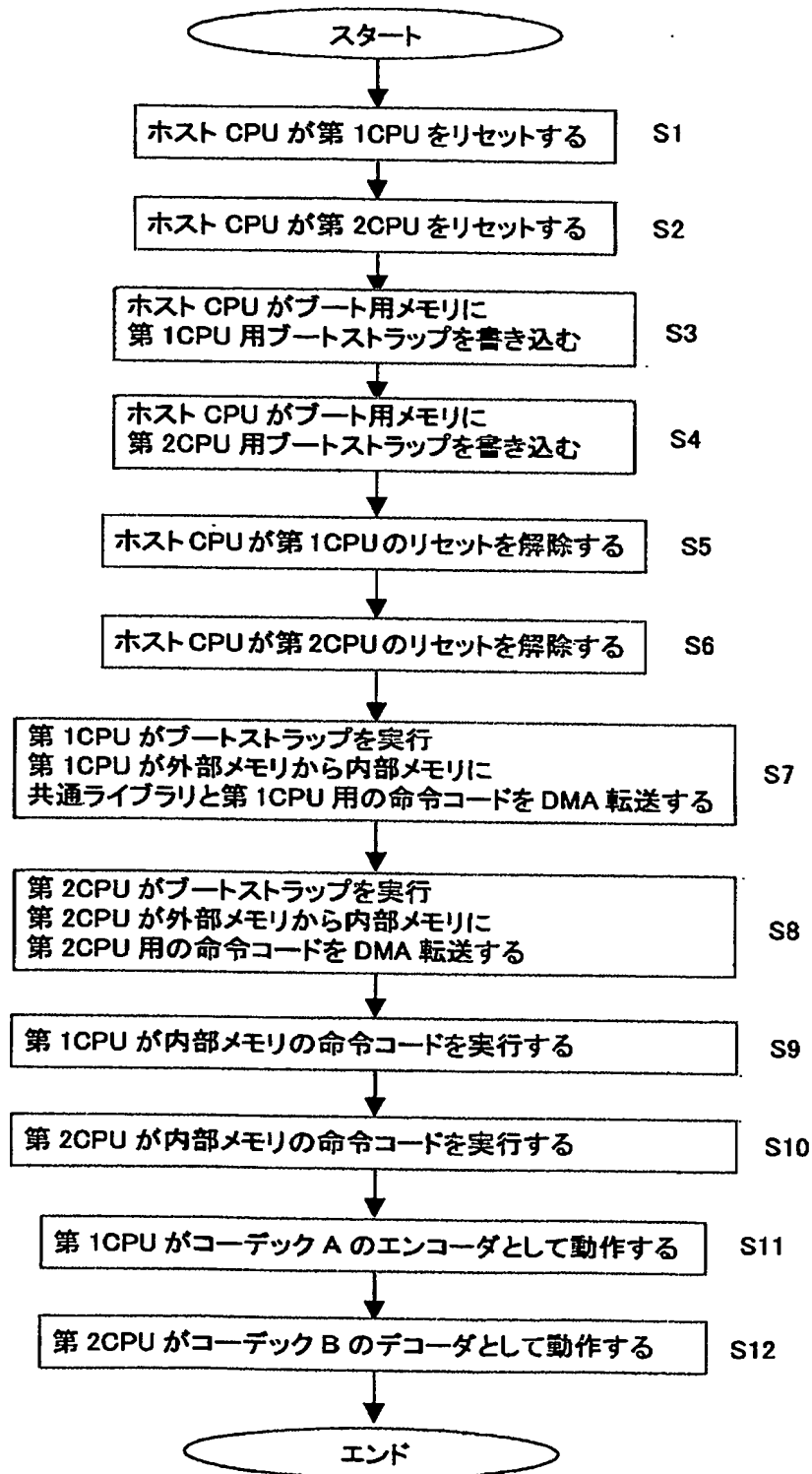




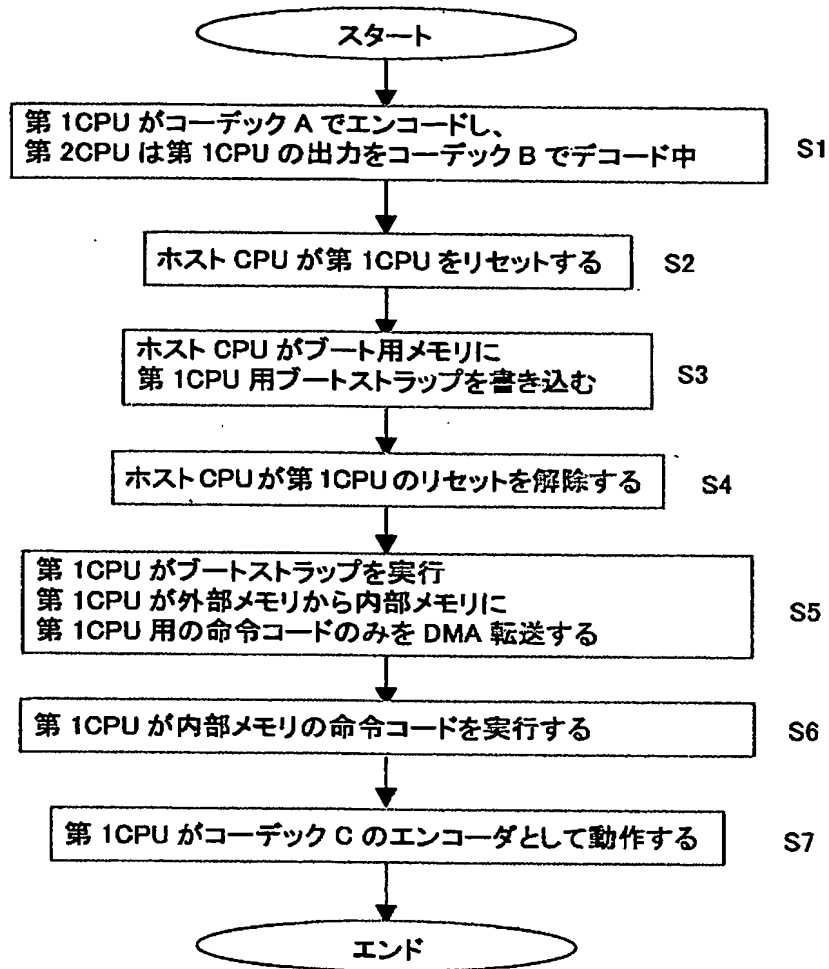
【図 2】



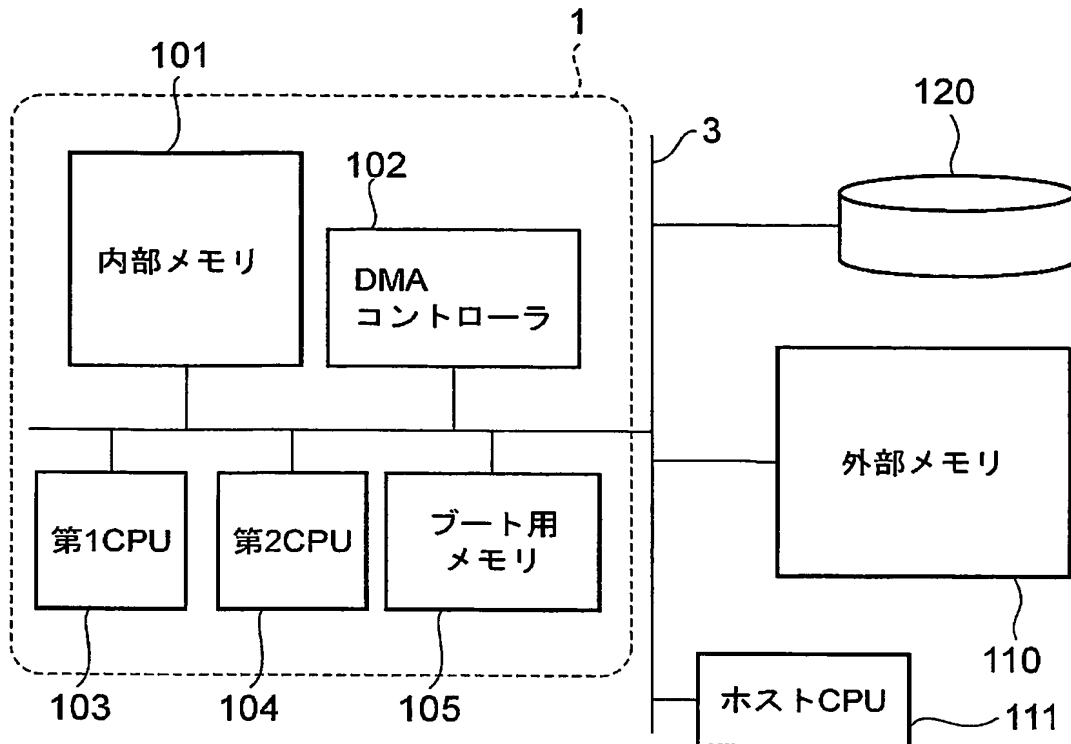
【図 3】



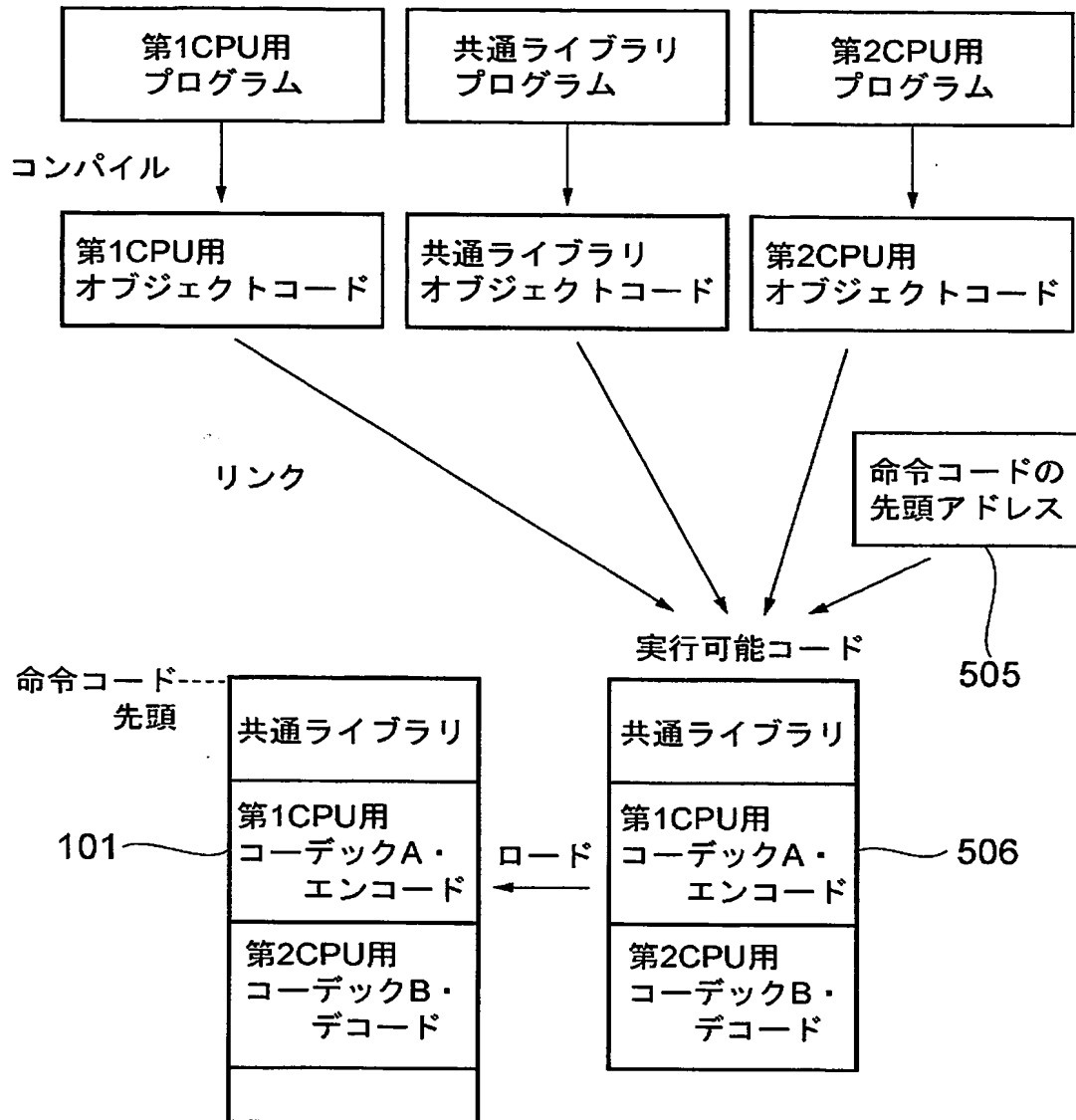
【図 4】



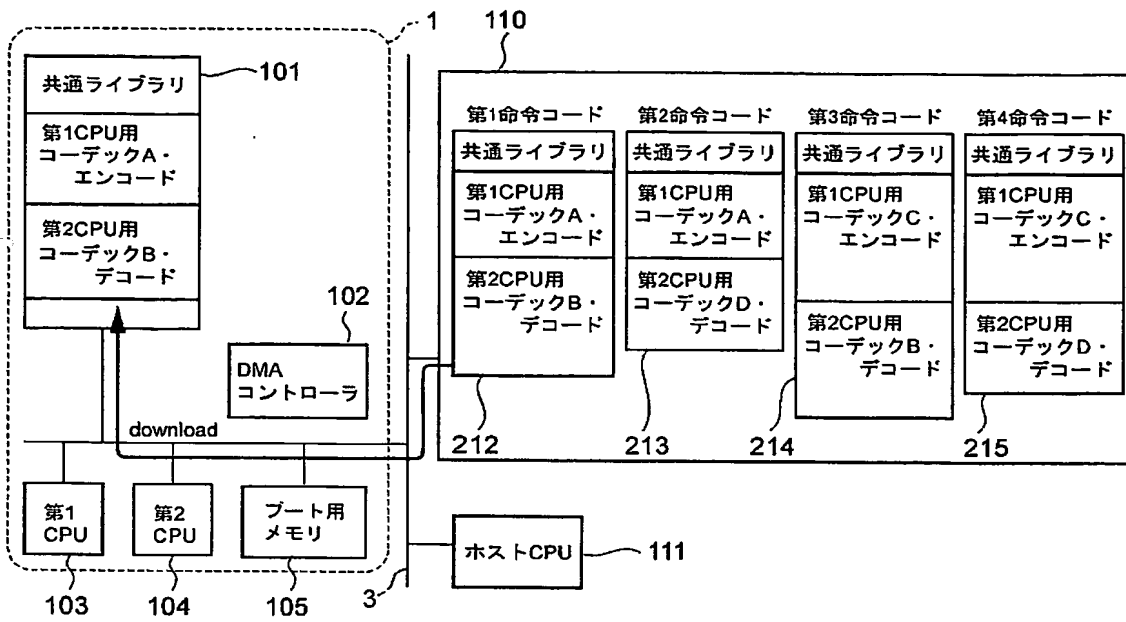
【図 5】



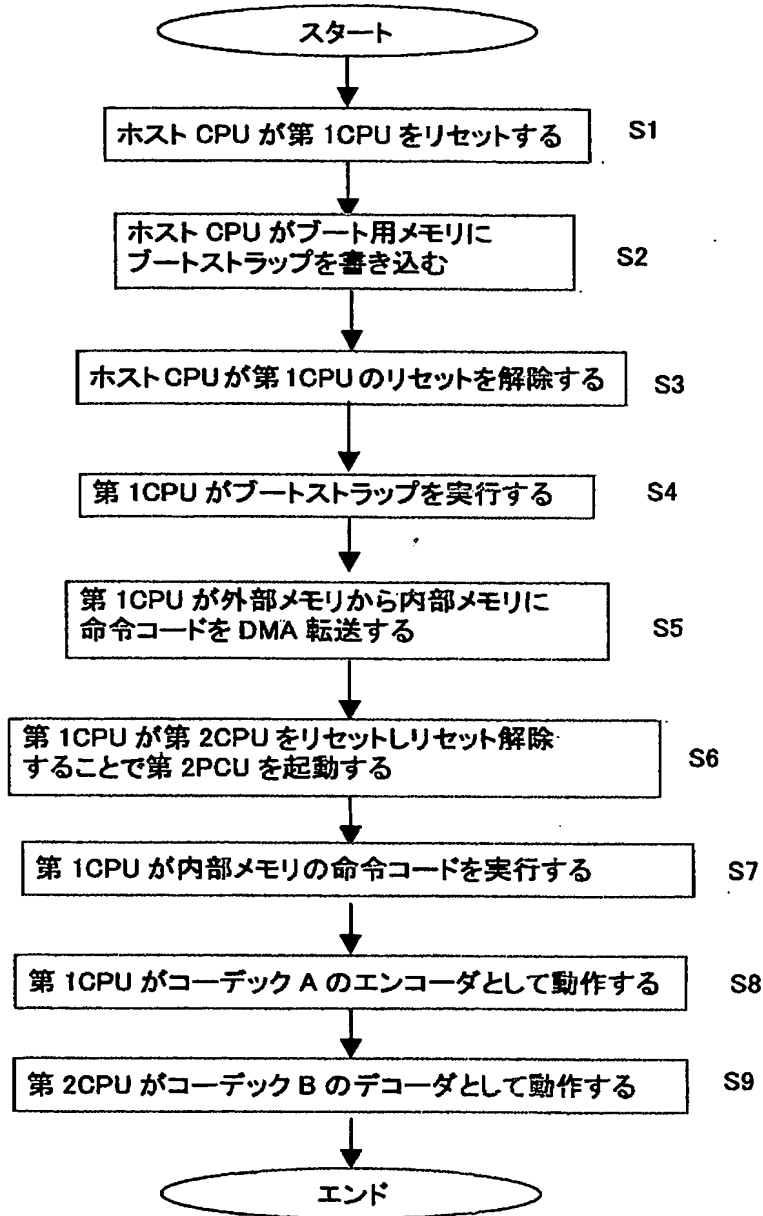
【図 6】



【図 7】



【図 8】



【書類名】 要約書

【要約】

【課題】 複数のCPUを有するシステムにおいて必要とされる記憶容量を低減できると共に処理速度を高速化し、他のCPUの動作に影響を与えることなく各CPUの機能を容易に変更することのできる情報処理方法を提供する。

【解決手段】 ホストCPU 411の指示により、いずれか一つのCPU 103, 104が共通コードと自己において実行するよう定められた命令コードとを外部メモリ 110から内部メモリ 101へロードすると共に、他のCPUが自己において実行するよう定められた命令コードを外部メモリ 110から内部メモリ 101へロードし、各々のCPUが、内部メモリ 101にロードされた自己において実行するよう定められた命令コードを実行すると共に、内部メモリ 101にロードされた共通コードを必要に応じて実行することを特徴とする情報処理方法を提供する。

【選択図】 図1



特願 2002-238437

出 願 人 履 歴 情 報

識別番号

[000002185]

1. 変更年月日

1990年 8月30日

[変更理由]

新規登録

住 所

東京都品川区北品川6丁目7番35号

氏 名

ソニー株式会社